Semantic Segmentation Using Hybrid Markov Logic Networks

Aravindh Mahendran
amahend1@andrew.cmu.edu

Nitish Thatte
nitisht@andrew.cmu.edu

Adwait Gandhe
agandhe@andrew.cmu.edu

Abstract

In this paper, we propose a new approach for Semantic Segmentation using Hybrid Markov Logic Networks.

1 Introduction

2 Related Work

3 Method

3.1 Label Propagation

In the related works section above, we discussed a method of propagating labels on a graph of connected superpixels based on a learned label transfer confidence function. This method introduced us to two ideas: propagating labels rather than directly minimizing a cost function to arrive at a segmentation and working with graphs of related superpixels. Building on these ideas, one approach we decided to pursue is to learn the weights for a hybrid Markov logic network in order to refine a prior expected segmentation obtained from a baseline algorithm through maximum a posteriori estimation. The method incorporates an undirected graph of connected superpixels that expresses where labels can propagate.

Examples of numeric terms and predicates in this Markov logic network are:

$$featureDistance(s_i, s_j)$$
$$isLabel(s_i, l_k)$$
$$isNeighbor(s_i, s_j)$$

Where s_i is the ith superpixel in an image and l_k is the kth label. $featureDistance()$ is a numeric term that returns the computed distance between two image feature vectors via a distance metric such as euclidean or cosine distance. $isLabel()$ returns 1 if superpixel s_i has label l_j and 0 otherwise, and $isNeighbor$ returns 1 if its arguments are the neighbors and 0 otherwise.

An example of a formula in this hybrid Markov logic network that use these predicates and numeric terms are :

$$[isNeighbor(s_i, s_j) \Rightarrow (isLabel(s_i, l_k) \Leftrightarrow isLabel(s_k, l_m))] * (featureDistance(s_i, s_k, f_m))$$

This formula encodes the following two beliefs about how and why labels can propagate in the image.

1. Superpixels with similar feature vectors should have the same label.

2. Superpixels that are connected by an edge (are neighbors) should have the same label.

A brute-force approach to this problem would be to allow every super pixel to transfer its label to every other super-pixel. This would result in a complete graph, where there is an edge between every two distinct nodes. Such a graph would allow superpixel labels to be learned from a more global context in the image. However, there would be an immense number of formulas for which to learn weights, making finding the solution computationally intractable.

In contrast, a simple approach, which has been attempted in the past, [] is to connect nodes that are directly share borders. Consequently, the graph would very few edges, greatly reducing the number of required learned weights and simplifying the learning problem. This approach, however, would only use local context for label propagation.

Therefore, we propose to take a middle ground approach that will produce densely connected superpixels where relations are expected to be strong, and fewer edges across boundaries expected to not share labels. In order build this superpixel graph, we take two levels of superpixel segmentations: A coarse level, which we will henceforth refer to as supersuperpixels, and a fine level, termed superpixels. All superpixels within a supersuperpixel are connected allowing for contextual information to be shared within a region. Additionally superpixels that share a boundary between two supersuperpixels are connected allowing for labels to propagate across supersuperpixels while acknowledging that label transfer across supersuperpixels is less probable than within a supersuperpixel.

4 Experiment

5 Conclusion

6 Design of Upcoming Experiments

7 Plan of Activities
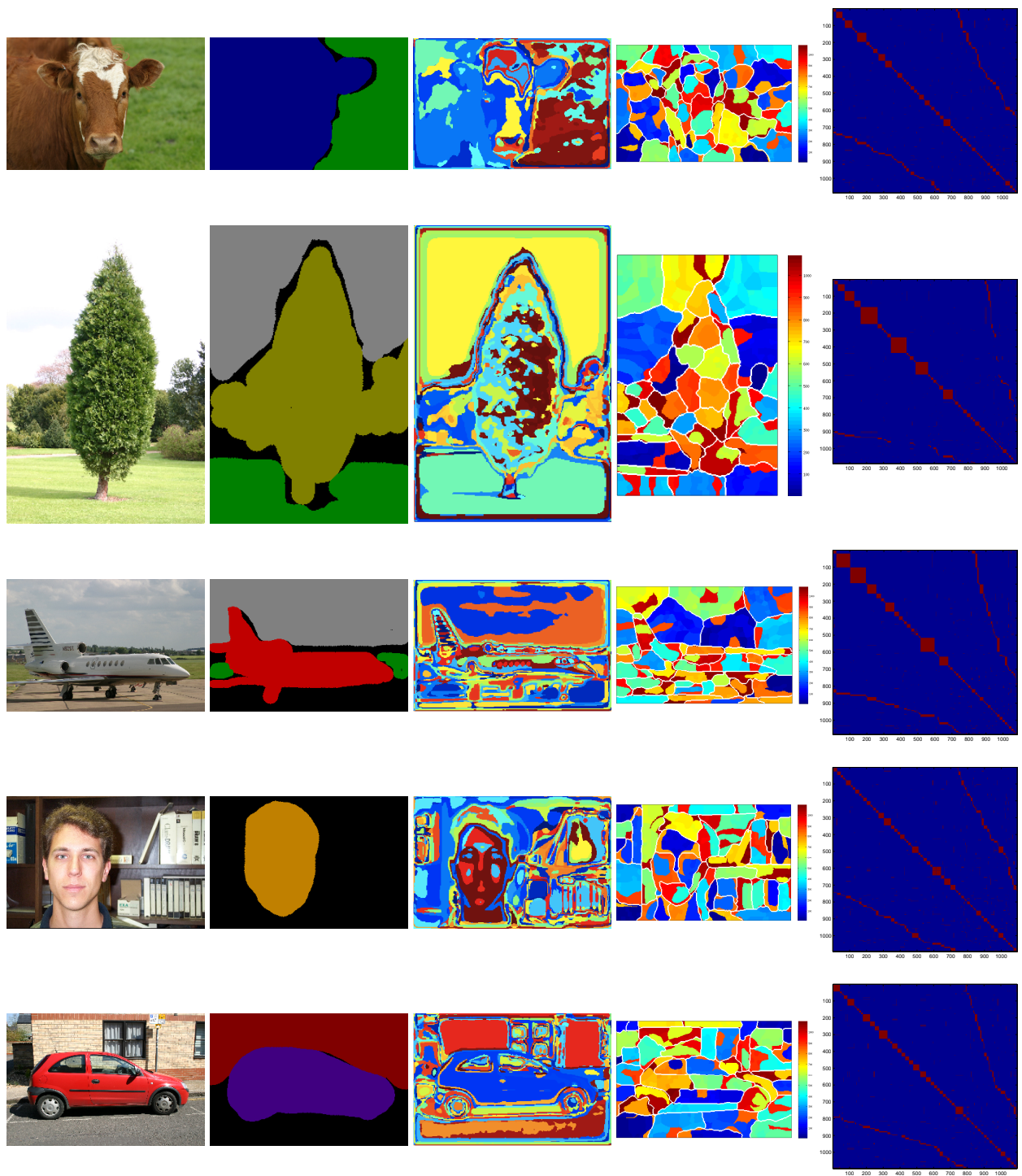
(a) original image   (b) ground truth segmentation   (c) visual words map   (d) superpixel segmentation   (e) superpixel adjacency matrix

Figure 1